Everyday-Organiser

Mathew Wilson

COLLABORATORS

	TITLE : Everyday-Organiser		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Mathew Wilson	August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Everyday-Organiser

1.1 The Everyday Organiser System.

The Everyday Organiser System. v2.00

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1.2 Introduction to the Everyday-Organiser.

Everyday Organiser (EO) consists of three main programs designed to help you manage your personal day-to-day activities. You can store friends addresses while keep up-to-date on events such as birthdays and appointments etc. It acts slightly differently to some other systems but is easy to use.

· The functioning of the Organiser system is split up into different programs so you may choose which ones to use, and how to use them.

· The main programs are supported with a large range of ARexx commands to help solve those difficult situations.

· You will need Magic User Interface (MUI) v3.3 to operate the main programs.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The programs included with EO are:

Locate....: Yet another contestant in the wonderful world of Address Management

programs ;-). It has a few options that separate it from the rest.

Alarmist..: A daily alarm management program that can be used on its own but is

best used in conjunction with On-Time.

On-Time...: A personal event and task manager.

There are four extra utility programs included:

TimePrefs.: MUI replacement for the Workbench Time preferences program that

directly supports Alarmist and On-Time.

Talk......: Replacement for CLI 'Say' command. Has quite a lot more features!

Sam.......: Small & simple yet useful CLI sound-datatype player.

Coder.....: Small CLI program to encrypt files.

I hope that Everyday Organiser does all that you need and is fun to use,

Mathew Wilson - June 1996, Adelaide, South Australia.

1.3 Notes on Installation

You may install or update EO by double-clicking on the 'Install_EO' icon (you will need the standard Installer program).

The installer will copy all the EO program documentation files to the Everyday:docs/<language> drawer. You should leave them there if you want on-line help.

You may be asked by the Installer whether you wish Alarmist to be copied into your WBStartup drawer. This is because Alarmist is required to act as a commodity and be present at all times. It doesn't make much sense to put it anywhere else.

A file called "Talk_Defaults" will be copied to your "S:" directory. It is an optional file used by the program "Talk" located in the Extras drawer. Consult the Talk documentation to find out more on this file.

After installation you'll probably end up with duplicates of some of the programs because there were copied from the installed drawer to other places like C:. The actual location of all the individual programs is not important, so you may reorganise things to how you like them best. Just remember to read the program docs to find out other relevant files.

The installer will make the assign "Everyday:" to the EverydayOrganiser drawer.

If you want to do the installation by hand then just drag the EO sub-drawer onto your HD. Make sure you copy the LIBS: drawer over to the correct place too! Within the Extras drawer you should copy Talk, Sam and Coder to C: and TimePrefs to your Preferences drawer. Finally if you install by hand, you are going to need to add the "Everyday:" assign to your S:User-Startup for the EverydayOrganiser drawer and copy the catalogs to their respective locale drawers. Then make yourself a coffee and read the docs.

1.4 Conversion

Converting data files.

· Locate

EO 1.0 and EO 1.1 Locate data files will be 100% compatible with EO 2.0.

· On-Time

EO 2.0 contains a small program for converting EO 1.0 and EO 1.1 On-Time data files. Run the program from workbench and it will convert ENV:OnTime.list; otherwise run from shell and specify new file. Remember to run the On-Time after converting ENV:OnTime.list and select SAVE.

· Alarmist

E0 1.0 and E0 1.1 files will be compatible BUT you must make one small manual change. Open a text editor (e.g. c:ed) and open ENVARC:Alarmist.list. You should find the first two lines contain:

- "ALARMIST"

- "5"

We need to make a small adjustment to the third line. It will contain a single number. You must add one (1) to the number and replace it with this new value. So if it read "-1" it will now read "1", "0" will now read "1" etc. Make the alteration and save the file. If you are not going to reboot then save the file to the ENV: drawer as well. The file version (5) was not changed due to the programmer not getting bored enough 8).

· Tasker

The EO 1.1 file can be converted to an On-Time 2.00 file with the supplied Tasker converter program. Once a file is converted it *MUST* be MERGED into On-Time via the Project menu Merge Tasks function. Then you'll have all the Tasks appear on the Task page of On-Time (no filter). Now to keep these Tasks just hit SAVE from the Main Window.

The program Tasker is now obsolete and will NOT function properly with the EO 2.00 system. I suggest you delete it, along with its docs that may may been copied into HELP:.

1.5 Importing and Exporting

Importing and Exporting

· Locate

Locate contains import facilities. You may export data via the Print Data function using "Columns" which separate the fields with TABS. See program documentation for further explanation.

· Alarmist and On-Time

These programs contain no direct importing or exporting functions. This was left out because any importing and exporting functions would be too program specific and not very useful or flexible. BUT, and I repeat BUT (several million times 8) you may use the program's ARexx interface to accomplish the same thing and tailor the output/input your own situation. For example you may have a personal electronic organiser that interfaces with the computer. I have no such system (no money - unemployed) but you could transfer data using an ARexx script. The programs (including Locate) have the two ARexx commands GetField and SetField which you can use to transfer data between EO and your electronic organiser. This way the data would be transferred in the correct manner. It would be impossible to build a function into the programs that did this for all organisers on the market (unless there is a standard - which I wouldn't be aware of). As usual, the good old Amiga come to the rescue again with ARexx.

1.6 Languages

Language Implementation for Everyday Organiser

As of version 2.00 (EO), all MUI programs have been completely cataloged. When you run the installer it will copy the language catalogs of your choice to your hard-disk.

When you run On-Time in a supported language other than English you'll notice that some of the user-configurable strings are still in English, e.g. Custom Filter strings. This is because all the global settings (except Events, Tasks and Notes) are saved to the program's icon. You can of course alter these strings to your own language in the Configuration. For the Filter String commands click on the button marked "?".

If there is no catalog for your spoken language and you wish to translate the EO catalogs, then please email me regarding this and I will send you the catalog translation files. I have not distributed them with EO because I didn't want two or more people translating into the same language and ending up with slightly different duplicate catalogs. I would be grateful for your assistance in this area. All new Catalogs are to be distributed with future versions of EO.

The EO documentation is currently available only in English. There are two reasons for this: firstly, it is constantly changing and secondly, it would take quite a while to translate. Then there is the size consideration. However, the MUI programs will automatically look at your Locale preferred language settings and look for the respective documentation.

If you change the system Locale Preference settings without rebooting, EO will not automatically update to reflect this change. Apart from the fact that you always need to restart programs when you change the Locale settings, the everyday.library needs to be flushed from memory. So first, quit any program that uses the everyday.library (i.e. Alarmist, On-Time, Locate, TimePrefs), then type "avail flush" from a shell. The library will be flushed and you may run the programs as you normally do. Of course it would have been a lot simpler just to reboot, but if you have a reboot time like mine then this might get a little boring 8).

1.7 Compilation Details

Compilation Details

Everyday Organiser was written using SAS/C version 6.55.

The programs were compiled with the following options:

- String Merge
- Standard Maths
- All CPUs
- Parameters passed on registers
- No Stack Checking
- Inline calls to ROM Utility.Library

All programs were compiled with optimisation options:

- Global Optimiser on
- Peephole Optimiser on
- Instruction Scheduler on
- Loop Optimisation on
- Optimisation for Time not Size

1.8 Why not register today

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With

the aid of a preferences program, the user of an application has the

ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing

lots of examples and more information about registration please look for

a file called "muiXXusr.lha" (XX means the latest version number) on

your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.9 thoughts

Definition: "How to make a programmer into a nervous wreck!"

When programming with MUI 3.3 I have come across something bizarre - twice. All my sub-windows inherit the Application Menu. I might use the following notification for a menu:

```
DoMethod(FIND_MENU(Menustrip, Menu_ConfigLoad), MUIM_Notify,
MUIA_Menuitem_Trigger, MUIV_EveryTime,
obj, 2, MUIM_MainWindow_ConfigLoad, ProgName);
```

Where 'obj' is the Main Application Window. Now, with SOME of these menus, when they are called while a different window is actvated, the program will crash. BUT, if the Main App Window is active then everything goes fine! Many, many wreckless hours of nervously twitching everytime it happens have been spent trying to figure this one out. I can't give a detailed description of the "crash" because that's what I spent all those hours trying to put my finger on. Just when I thought I had it figured out, it would crash differently. I came up with a fix but I don't understand why it should fix it. If you can help me I would be very happy. Am I doing something wrong? My solution to the problem is to use a Push Method, e.g.:

```
DoMethod(FIND_MENU(Menustrip, Menu_ConfigLoad), MUIM_Notify,
MUIA_Menuitem_Trigger, MUIV_EveryTime,
MUIV_Notify_Application, 5,
MUIM_Application_PushMethod, obj, 2, MUIM_MainWindow_ConfigLoad, ProgName);
```

==

In all the MUI programs I have seen coming out there seems to be some lack of consistency in interface design. I would like to make a few suggestions. These are only my personal opinions and not necessarily the opinions of others including Stefan Stuntz. I welcome comments and suggestions. I have no money for Internet access on a regular basis so subscription to the MUI Mailing list is out of the question for now.

· Wherever there is an instance of a gadget that has an associated label alongside of it, I would like to see them both enclosed in a group frame and group background. My reasoning for this is that often users have complicated window backgrounds and text wacked straight on top can be unreadable. If you place all your label objects on a group background WITH a group frame them the user can select an appropriate group backdrop to increase readability. There will be instances when this is silly or unavoidable. A single gadget with a label in a window full of non-labeled gadgets may look out of place.

I feel programmers should be more careful with the type of frame and background they choose. Thinking twice before selecting a frame will aid in maintaining a consistent look for all applications. I have seen sets of text objects grouped together in a text frame containing a window background. This doesn't make much sense to me at all. In such case, the programmer might want a text frame but surely you should use a group background. Another example is a radio gadget smacked straight on top of a window - very hard to read indeed!

I think group frames and group backgrounds should be used more often (within reason). I use the following definition when I want to enclose or group together a set of objects.

```
#define MyGroupFrame GroupFrame, MUIA_Background, MUII_GroupBack
#define MyGroupFrameT(s) GroupFrameT(s), MUIA_Background, MUII_GroupBack
```

· Menu layout on the Amiga is almost a case of pot-luck sometimes. Other OS platforms have rigid layout rules but don't provide for flexibility. Running through menus to find a particular item can be a real pain sometimes. I believe that Starting with Project, Edit and Settings Menus is a good basis for logical menu layout. Throwing everything in the Project menu can be very confusing as well.

1.10 theauthor

The Everyday Organiser system was written by Mathew Wilson.

This project has taken a considerable amount of time and energy. I maintain it as Freeware in some distant hope that it might help the Amiga in some small way. I hope that this Freeware status does not offend any other programmers in anyway.

If you are an artist and would like to make some icons for EO then please send them to me for possible inclusion in the next release!

You may contact me (or send postcards) at:

Box 6022, Halifax Street Post Office

Adelaide SA, 5000

AUSTRALIA

I can be e-mailed via David Cottrell.

(I'm unemployed and can't afford my own account.)

dcottrel@adam.com.au

1.11 Thanks to...

Thanks to...

· Thanks to Dave Cottrell for running EO on his machine and letting me know when strange things occur. He has spent a great amount of time sending email and general stuffing around on Internet for me. Those email replies would not have been possible without the help of Dave!

· I started programming with C in April '94 and I would like to thank Peter Jackson for his invaluable help with learning C on the Amiga. He answered my many questions and gave me some great ideas along the way.

· A big thanks to Stefan Stunz for creating and maintaining the MUI system.

· EO v2.0 was my first fully Cataloged project (MUI programs only). The patience and suggestions of the following translators was much appreciated.

- Italian translator: Luca Danelon <danelon@interlandsrl.it>.

- German translator: Michael Zielinski <m.zielinski@amtrash.comlink.de>.

· Thanks to Kai Hofmann for the following:

- Week of year calculations are based on information provided by his 'date.lbrary'.

- Moon phase calculations use the formula found in his 'date.library'.

Kai has developed a library full of date calculating functions: 'date.lbrary' Aminet:dev/c/date.lha.

Greetings and thanks to those who sent me mail regarding EO. Feedback is always a good thing and it's nice to know there are other Amiga users on the planet 8). I appreciate it. It also helps to keep me from changing computer platforms 8).

I have received one postcard - from Germany "Rheinstetten". Thanks to R.E. Spence.

Thank YOU for taking the time to read the docs.

Unrelated Thankyous

Thanks to companies like Softwood and Phase 5 for their excellent support of the Amiga!

Thanks to all those who contribute software to help maintain the existance of the Amiga. One day the Amiga really will be a viable alternative! It's only a matter of hardware, software and time 8).

1.12 history

History

Changes to individual programs can be found by reading the "History" sections in their Amiga-Guide documentation found in the EverydayOrganiser/docs/... drawer.

· v2.00 - Third Release (8th June 1996)

Say goodbye to the program Tasker (from EO-Second release). Task management has been taken up by On-Time. In fact On-Time has been completely recoded from scratch! It is now radically different from previous versions and I recommend strongly that you re-read its documentation or your going to miss out on quite a lot of necessary details! I don't receive any financial rewards for the development of EO so you could at least do me the favour of reading through the docs. I have put an incredible amount of time into this project!

I was never really happy with On-Time v1.x. It contained severe limitations that continually annoyed me to no end. So I decided to recode it from scratch. I am now happy with it and can say that it does what I originally intended and hoped for. I hope the changes meet with your approval.

Major Changes Include:

- The "everyday.library" has been changed and bumped up to lib version 3.00.

- The Everyday: assignment was made. I avoided making this step for as long as I could, but now with language support and a few other things, I found it necessary.

- The programs now look for their documentation in the "Everyday:docs/<language>/" drawer. The Locale language preference list is used to select the appropriate file.

- MUI programs are now completely cataloged.

- Added "STARTUP_SCRIPT" tooltype for Alarmist, OnTime and Locate. Executes an ARexx script at run-time.

- 'Talk' now has full extended Narrator support.

- In previous versions, some windows used to make a string gadget active upon opening the window. This behaviour has been temporarily suspended because, believe it or not, strange things were occurring, including inability to snapshot the window and sometimes crashing. I dislike to hastily lay the blame with MUI but I can't think of anything else.

· v1.10 - Second Release (27th March 1996)

A new program called Tasker has been added to the system. It is designed to handle tasks/events that occur over a period of time. It might have been a little more logical to incorporate these facilities into On-Time instead of creating a new program. This may occur in the future (if I stay with Amiga).

Major Changes Include:

- The day after (true!) the first release I found a thousand and one Enforcer Hits. For this I am deeply sorry. There is no excuse for such criminal activities! I normally test for such nasties but this time I didn't! The problem was that I did an enormous amount of re-coding in the week previous to release. I tested most things except for the dreaded Enforcer Hit. Please accept my deepest apologies.

- A nasty memory/task allocation bug would occur EVERY time one of the programs sent an ARexx message. In such cases a console would be allocated and NOT de-allocated so you would end up with a console task hanging around after an ARexx message was sent. (not too good eh? oops 8).

- A new Tool-Type called 'DROP_SCRIPT' has been added to the main programs. This is not editable within the programs and must be set manually from the Workbench Information window. It allows you to specify where the Application Icon Drop Script (AIDS 8) is to be found. This allows you to finally put all the supplied ARexx scripts in a drawer of their own instead of lumping them all in the REXX: drawer (A practice that I'm not too fond of). The installation script creates a "Rexx" drawer in the "EverydayOrganiser" drawer for this purpose. So if you installed version 1.0 of EO then you may either choose to copy these new scripts to the REXX: drawer or delete the old ones in REXX: and use the new location.

- The extra program Sam was not fully documented in the first release.

- A Locate example file was added (in LocateData drawer).

- The "everyday.library" has been changed and bumped up to lib version 2.00.

- Some people reported severe problems with getting the main programs to work with the first release. EO has been running on my A2000, a friends A3000 and A1200 for a while with no problems. A friend was having a few list problems with Alarmist on his A4000 though. Hopefully this release fixes any problems. I apologise for any inconveniences.

· v1.0 - Original Release (8th February 1996)